

Programming for Data Science

How to create a package in R

Marco Beccuti

Università degli Studi di Torino

Dipartimento di Informatica



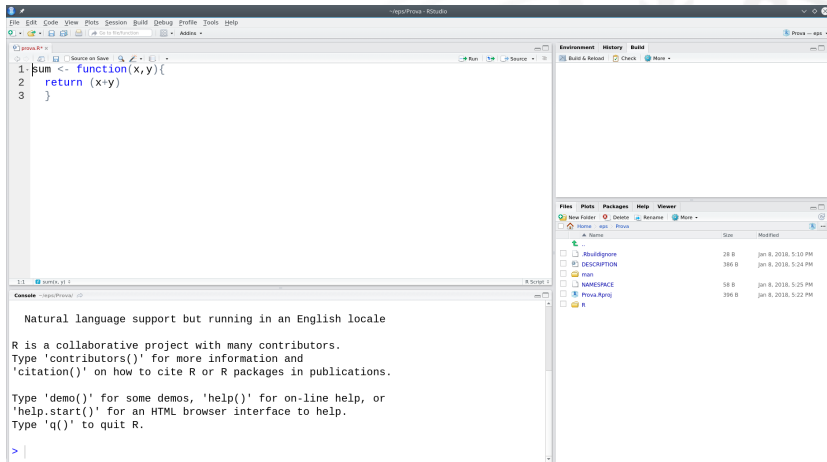
How to create a R package

- we describe how to create an R package using RStudio;
- Obviously it is possible to build R Packages with Command Line Tools, but RStudio has built-in tools that will make easier this task;
- the **devtools** package must be installed in RStudio.

```
>install.packages("devtools")
```

How to create a R package

- Open a new .R file;



The screenshot shows the RStudio interface. The main editor window contains the following R code:

```
1 sum <- function(x,y){  
2   return (x+y)  
3 }
```

The console window at the bottom shows the following output:

```
Natural language support but running in an English locale  
  
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
  
> |
```

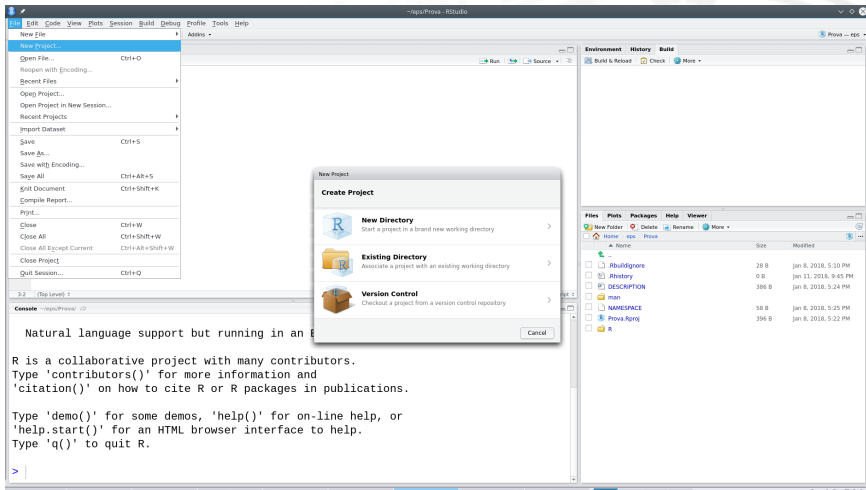
The right-hand side of the interface shows the 'Environment' and 'Files' panels. The 'Files' panel displays a directory listing:

Name	Size	Modified
..		
.._		
._buildignore	28 B	Jan 8, 2018, 5:10 PM
DESCRIPTION	386 B	Jan 8, 2018, 5:24 PM
man		
NAMESPACE	58 B	Jan 8, 2018, 5:25 PM
Prova.Rproj		
R	398 B	Jan 8, 2018, 5:22 PM

- Write the code for your functions in this .R file. You can create one file with all of your functions or create separate files for each function. Save these files somewhere where you can easily find them.

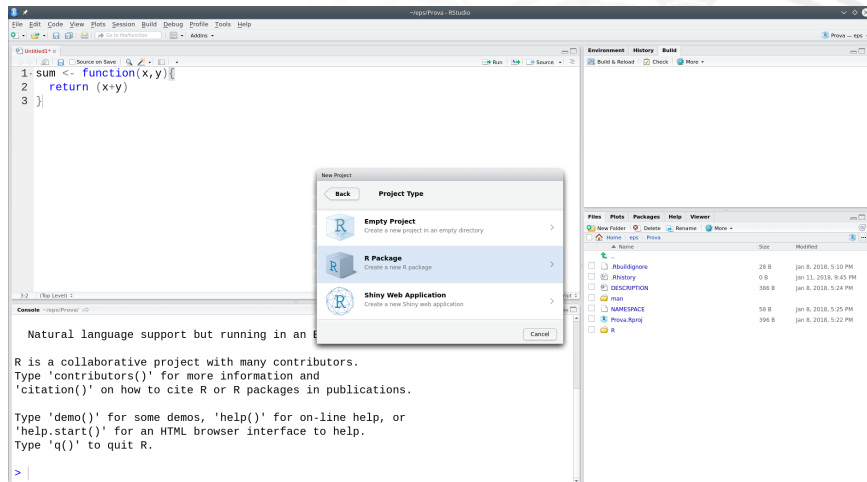
How to create a R package

- Open a new project in RStudio. Go to the **File** menu and click on **New Project**. Then select **New Directory**, and **R Package** to create a new R package.



How to create a R package

- Open a new project in RStudio. Go to the **File** menu and click on **New Project**. Then select **New Directory**, and **R Package** to create a new R package.



The screenshot shows the RStudio interface. The main editor window contains the following R code:

```
1 sum <- function(x,y){  
2   return(x+y)  
3 }
```

The 'New Project' dialog box is open, showing three options:

- Empty Project: Create a new project in an empty directory
- R Package**: Create a new R package (highlighted)
- Shiny Web Application: Create a new Shiny web application

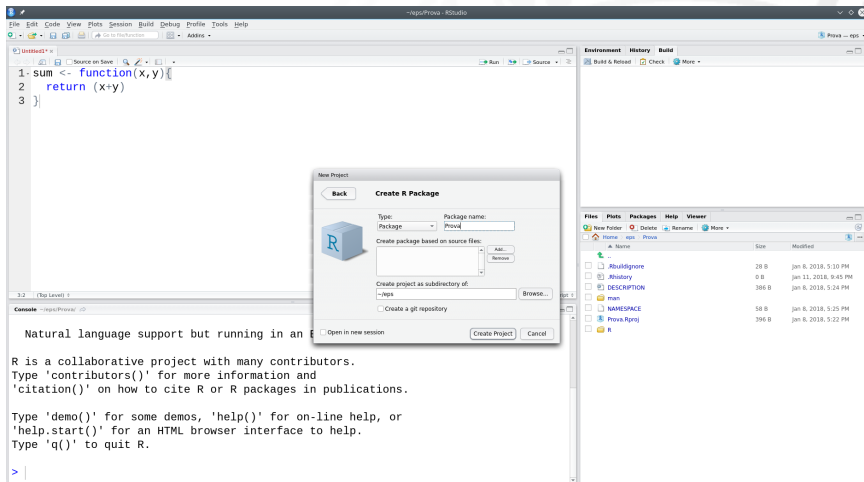
The console window at the bottom displays the following text:

```
Natural language support but running in an R session.  
R is a collaborative project with many contributors.  
Type 'contributors()' for more information and  
'citation()' on how to cite R or R packages in publications.  
  
Type 'demo()' for some demos, 'help()' for on-line help, or  
'help.start()' for an HTML browser interface to help.  
Type 'q()' to quit R.  
> |
```

The Environment pane on the right shows a file explorer view of the current project directory, listing files such as .Rbuildignore, .Rhistory, DESCRIPTION, main, NAMESPACE, Prova.Rproj, and R.

How to create a R package

- Type the name of your package, then upload the .R file you created in step 1 under **Create package based on source files**. Click **Create project**.



```
1: sum <- function(x,y){
2:   return (x+y)
3: }
```

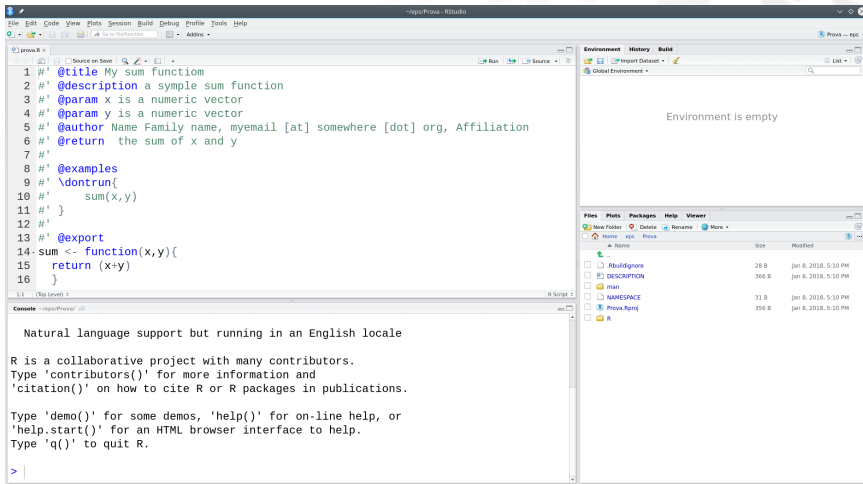
Natural language support but running in an...

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

How to create a R package

- Documentation can be automatically created using **roxygen2** package.



The screenshot shows the RStudio interface. The script editor contains the following code:

```
1 #' @title My sum function
2 #' @description a symple sum function
3 #' @param x is a numeric vector
4 #' @param y is a numeric vector
5 #' @author Name Family name, myemail [at] somewhere [dot] org, Affiliation
6 #' @return the sum of x and y
7 #'
8 #' @examples
9 #' \dontrun{
10 #'   sum(x,y)
11 #' }
12 #'
13 #' @export
14 sum <- function(x,y){
15   return (x+y)
16 }
```

The console output shows the following text:

```
Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

> **install.packages("roxygen2")**

How to create a R package

- The functions must be commented using the following tags:

```
##' @title My sum functiom
##' @description a symple sum function
##' @param x is a numeric vector
##' @param y is a numeric vector
##' @author Name Family name, myemail [at] somewhere [dot] org, Affiliatio
##' @return the sum of x and y
##'
##' @examples
##' \dontrun{
##'   sum(x,y)
##' }
##'
##' @export
sum <- function(x,y){
  return (x+y)
}
```


How to create a R package

- To enable the generation of package documentation using **roxygen2**

The screenshot shows the RStudio interface with a script editor containing the following R code:

```
1 #' @title My sum function
2 #' @description a symple sum function
3 #' @param x is a numeric vector
4 #' @param y is a numeric vector
5 #' @author Name Family name, myemail [at] somewhere [dot] org, Affiliation
6 #' @return the sum of x and y
7 #'
8 #' @examples
9 #' \dontrun{
10 #'   sum(x,y)
11 #' }
12 #'
13 #' @export
14 sum <- function(x,y){
15   return (x+y)
16 }
```

The console window displays the output of the `sum` function:

```
Natural language support but running in an
R is a collaborative project with many contri
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.
some demos, 'help()' for on-line help, or
an HTML browser interface to help.
R.
```

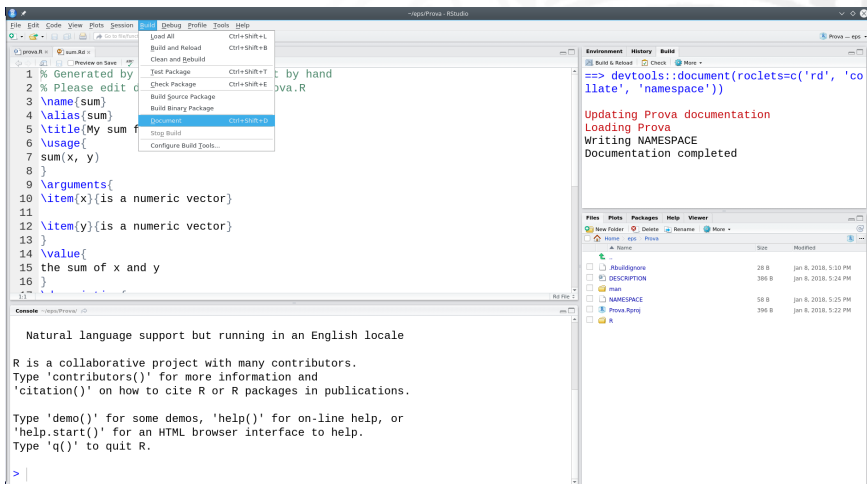
Two dialog boxes are open:

- Project Options**: Shows "Project build tools: Package" and "Package directory: [Project root]". The "Build Tools" section is expanded, showing options for "Build and Reload", "Check Package", "Build Source Package", and "Build Binary Package".
- Roxygen Options**: Shows "Use roxygen to generate:" with checked options for "Rd files", "Collate file", "NAMESPACE file", and "Vignettes".

A "New Project..." dialog box is also visible in the background, showing a list of projects to open or create.

How to create a R package

- Documentation is generated by clicking on **Build** → **Documentation**



The screenshot shows the RStudio interface with the 'Build' menu open and 'Documentation' selected. The console output shows the following steps:

```
==> devtools::document(roclets=c('rd', 'collate', 'namespace'))

Updating Prova documentation
Loading Prova
Writing NAMESPACE
Documentation completed
```

The console also displays the following text:

```
Natural language support but running in an English locale

R is a collaborative project with many contributors.
Type 'contributors()' for more information and
'citation()' on how to cite R or R packages in publications.

Type 'demo()' for some demos, 'help()' for on-line help, or
'help.start()' for an HTML browser interface to help.
Type 'q()' to quit R.

> |
```

How to create a R package

- Package can be built and loaded by clicking on **Build** → **Build and Reload**

The screenshot shows the RStudio interface with the 'Build' menu open. The 'Build and Reload' option is selected. The source file 'Prova.R' contains the following code:

```
1 % Generated by hand
2 % Please edit carefully
3 \name{sum}
4 \alias{sum}
5 \title{My sum function}
6 \usage{
7   sum(x, y)
8 }
9 \arguments{
10  \item{x}{is a numeric vector}
11
12  \item{y}{is a numeric vector}
13 }
14 \value{
15   the sum of x and y
16 }
```

The console shows the following output:

```
Restarting R session...
> library(Prova)
Attaching package: 'Prova'
The following object is masked from 'package:base':
  sum
> ?sum
>
```

The Environment pane shows the package 'Prova' and its components:

Name	Size	Modified
..		
..buildignore	28 B	Jan 8, 2018, 5:10 PM
DESCRIPTION	396 B	Jan 8, 2018, 5:24 PM
man		
NAMESPACE	58 B	Jan 8, 2018, 5:25 PM
Prova.Rproj	396 B	Jan 8, 2018, 5:22 PM
R		

How to create a R package

- Package can be built and loaded by clicking on **Build** → **Build and Reload**

The screenshot displays the RStudio interface with the following components:

- Source Editor:** Contains the `DESCRIPTION` file for the 'Prova' package. The content is:

```
1 % Generated by roxygen2: do not edit by hand
2 % Please edit documentation in R/prova.R
3 \name{sum}
4 \alias{sum}
5 \title{My sum function}
6 \usage{
7   sum(x, y)
8 }
9 \arguments{
10  \item{x}{is a numeric vector}
11
12  \item{y}{is a numeric vector}
13 }
14 \value{
15   the sum of x and y
16 }
```
- Console:** Shows the execution of the following commands:

```
> library(Prova)
Attaching package: 'Prova'
The following object is masked from 'package:base':
  sum
> ?sum
> |
```
- Environment/Build Panel:** Shows the output of the `Build and Reload` action:

```
CONVERTING help for package 'Prova'
** preparing package for lazy loading
** help
*** installing help indices
    finding HTML links ...      sum
                                html
done
** building package indices
** testing if installed package can be lo
aded
* DONE (Prova)
```
- Viewer Panel:** Displays the help page for the `sum` function, including its description, usage, arguments, and value.